The SORCERER'S APPRENTICE

Vol. 1, No. 2

July, 1979

MISCELLANEOUS NOTES

As everyone probably noticed, the listing in issue #1 had a few
errors relating to lacking greater-than and less-than signs. Certain
lines in the program should be altered. I will use a "@" to represent
a greater-than sign and a "¢" for a less-than sign. Correct as follows:

```
210 IF LOC+64 @ -2049 THEN RETURN
310 IF LOC-1 ¢ -3968 THEN RETURN
410 IF LOC+1 @ -2049 THEN RETURN
510 IF LOC-64 ¢ -3968 THEN RETURN
```

Also pertaining to errors, several of you noticed that the correct
locations for the USR function addresses are not 259 and 260, but 260
and 261. These are the numbers given in the finished Sorcerer Technical
Manual, which is now in print and available for $11.95 from your local
computer store.

My thanks to Bob Bachman and his printer in Lansing, Michigan. Now
I will be able to provide clear, error-free listings.

CIRCULATION

My original ad in ON_LINE has only drawn about 30 responses so far,
althoughthey still keep coming in. The August issue of Kilobaud Micro-
computing has a letter that I sent in; this should generate some re-
sponse also. My aim is to reach all users, or at least to let them
be aware of our presence. One easy way of reaching others easily that
I have "discovered" is to place copies of this newsletter in computer
stores (with the owners permission, of course). Actually, this helps
us in more ways than one. Sorcerer owners will be pleased to see that

yes, some people are communicating ideas, and prospective buyers may
be more likely to purchase a Sorcerer if they know that an information
exchange does exist. Hopefully, both the user and buyer will contact
me. I have given copies to the stores in my area, so how about you?
You may well just come away from the store knowing more than when you
walked in (I certainly did), and friends in computer stores never hurt
anybody. If you think you can use some copies, whether to place in
a store, for friends or for that other computer club, let me know
how many you need and I'll get them out to you right away.

RUMORS AND NEWS ABOUT EXIDY

Do you still read ON_LINE? Then you probably already know about
Exidy's latest releases. If not, then you can read about them here.

Two new ROM PAC's are available- the WORD PROCESSOR and the DEVELOP-
MENT PAC, and there is a VIDEO/DISK attachment. The Video/Disk unit
houses a 12" video display using P31 phosphor and having a bandwidth
of 20 MHz for clear pictures. Also included is a dual mini-floppy
disk drive with a data storage capacity of 630K bytes. Software
included in the package includes CP/M, Z-80 assembler, text editor,
linking loader, and Microsoft Disk Extended BASIC. All for only $2995.

The Word Processing Pac will support either a modified Selectric or
the Diablo/Qume proportional output printers. Either a cassette recor-
der or a mini-floppy may be used for storage. The Edit mode allows
full cursor control, Insert/Delete, Scan, Tab, Indent, Hyphen, Macros,
pagination and titling. The Command mode has standard word processor
functions, tape merge with memory, line length set, printer option set,
string search, and display of unused space. File names may be up to 8
characters. Price $99.

Development PAC software includes a two pass Z-80 assembler, a
text editor, and a debugger. The assembler's I/O can be vectored to
any device driver within the Sorcerer, source and object code can be
spooled to accomodate programs of infinite length, and absolute assem-
blies and pseudo operators are supplied. The editor is line oriented,
allowing forward cursor positioning, line insert and delete, spooling,
and source code I/O to any device. The debugger can display and/or
modify any register or RAM location and set breakpoints. Available
for $99. For more info, contact Exidy directly.

Now for those rumors I promised. Exidy has not verified these, but I believe my sources to be reliable enough to print what follows.

An extended BASIC more powerful than Radio Shack's Level II will be available in the near future on cassette. This will be Microsoft's newest BASIC; it will include verbs to manipulate graphics, have long variable names, and other niceties.

The upcoming color graphics board will be bit by bit programmable and provide for a 256 by 256 display, expandable to 512 by 512. Note that even the low resolution mode has a higher resolution than the APPLE-II's high resolution mode. This should make for some beautiful displays.

PROGRAM EXCHANGE

No, not yet, but coming soon with your help. Everyone would be able to benefit from a cassette software exchange, and I'd like to see one going as soon as possible. I still want to publish listings; they are generally easier for me to follow than the video. Typing in lines is a real pain, however, and hard-to-find mistakes always seem to creep in. Microcomputing is supposed to be a fun hobby, not a tedious one. To that end, I think a free or inexpensive exchange should be set up. Notice please that I said that it should be set up, not that I can set it up. School still has first priority over my time, and most of my spare time goes to this newsletter in one way or another. I'd really like to hear from anyone else interested in performing this service. Who knows? You may even make some money doing it. Please, anybody with ideas, let me know what we can do.

WE ARE NOT ALONE....

In the user's group buisness, that is. Two other user's groups have come to my attention in recent weeks. One is sponsored by the Computer Mart of Waltham, Massachusets. You can write them at Computer Mart Waltham, 1395 Main Street, Waltham, Mass., 02154. I am sending a copy of this issue and a letter of introduction to them; I'll tell you more about their activities in the next issue.

The other group to come to my attention is run by Steve Long, 792 Laurie Ave., Santa Clara, CA, 95050. Steve publishes the Sorcerer User's Newsletter 12 times a year; subscriptions run for $10. Sample issues are available for the price of an SASE. Steve sent me a copy of

his May issue which included articles on a printer driver in Basic, some
digital to analog conversion techniques, and information regarding the
use and operation of some programs just added to his software library.
Steve's program already has 69 programs available for $4 apiece. He is
also a dealer for Exidy products.

FILES AND YOUR SORCERER

The Sorcerer's standard has provisions for saving, from within
a program, numerical arrays using CSAVE* and CLOAD*. These are great
if you want to write a checkbook handling program, but not much good
if you want to maintain an address list or phone book on tape. Of
course, since a computer can do virtually anything, saving and reading
character records (or arrays) is possible, but it has to be done in a
somewhat roundabout way. I have managed to _fairly_ successfully read
and save character strings using the ordinary PRINT and INPUT state-
ments after changing the input and output vectors from within a pro-
gram. This is similar to issuing the command SE O=S from within the
running program, and then resetting it with SE O=V, also from the
running program. Once again, I have used the USR function to accomplish
this, along with some simple machine language programming. I use
the following routines to change the output and input with the USR
function:

```
    Change output from video to tape-
    0000    11 12 EO    LD      DE,OUTAPE
    0003    FD 73 3F    LD      (IY+OUTADD),E
    0006    FD 72 40    LD      (IY+OUTADD+1),D
    0009    C9          RET

    Change output from tape to video-
    0010    11 1B EO    LD      DE,VIDEO
    0013    FD 73 3F    LD      (IY+OUTADD),E
    0016    FD 72 40    LD      (IY+OUTADD+1),D
    0019    C9          RET

    Change input from keyboard to tape-
    0020    11 OF EO    LD      DE,INTAPE
    0023    FD 73 41    LD      (IY+INADD),E
    0026    FD 72 42    LD      (IY+INADD+1),D
    0029    C9          RET

    Change input from tape to keyboard-
    0030    11 18 EO    LD      DE,KEYBRD
    0033    FD 73 41    LD      (IY+INADD),E
    0036    FD 72 42    LD      (IY+INADD+1),D
    0039    C9          RET
```

The following program uses the routines to print a character string to the tape unit and then read it back:

```
10 INPUT "TYPE A STRING";A$
20 PRINT:PRINT "PUT A CASSETTE IN THE RECORDER,
30 PRINT "START RECORDING AND ALLOW THE LEADER TO PASS
40 INPUT "HIT RETURN WHEN READY";XX$
50 POKE 260,0:POKE 261,0
60 XX=USR(0)
70 PRINT ""
80 PRINT A$
90 POKE 260,16
100 XX=USR(0)
110 CLEAR 50:REM NULL ALL VARIABLES
120 PRINT:PRINT "YOUR STRING HAS BEEN SAVED ON TAPE
130 PRINT "AND ITS VALUE IN THE VARIABLE A$ DESTROYED
140 PRINT:PRINT "TO RELOAD THE STRING, REWIND THE TAPE
150 PRINT "AND PRESS PLAY
160 INPUT "HIT RETURN WHEN READY";XX$
170 POKE 260,32:POKE 261,0
180 XX=USR(0)
190 INPUT XX$
200 INPUT A$
210 A$=RIGHT$(A$,LEN(A$)-1)
220 POKE 260,48
230 XX=USR(0)
240 PRINT:PRINT "THE STRING SAVED ON TAPE IS
250 PRINT A$
```

Line 70 is apparently necessary to initialize the INPUT statements so that the actual string is read back in instead of garbage on the tape. While in the input section of the program, you will see question marks printed on the screen as well as the garbage and the string itself. To prevent this from happening in a program, the output can be routed to the tape unit as well; this won't interfere with program performance, but will keep the question marks and garbage and strings from appearing on the screen.

Line 210 is used to chop off the first character of the INPUT string. For some reason unknown to me, the first character of the string is a CHR$(10), or a line-feed. I can only assume that this has to do with the way the PRINT statement operates.

I have used this I/O changing technique with some degree of success in a small address list handling program. Hopefully the program will be perfected in time for the next issue.

PROGRAMS OF THE MONTH

RACETRACK
contributed by Marvin Weingast

From Marvin: "This program is similar to the one published in
issue #1 except that the machine language subroutine, which is a little
different from yours, is inside a FOR-NEXT loop. This allows the mo-
ving spot, in this case a blinking star, to be moving all the time.
The rate of blinking or the speed of the star can be changed with mi-
nor programming changes.

"The object of the game is to maneuver the star around the track,
or alternatively to erase the track borcers using the star.

"To start the motion of the star press 2, 4, 6, or 8, which con-
trol the direction in the same manner as your program. Press 2, 4, 6,
or 8 to change the direction. To stop press 5."

LUNAR LANDER
contributed by Bob Bachman

Lunar Lander is one of the truly classic computer games, second
only to Startrek in use. Bob's version not only provides the captain
with the usual velocity, distance, and fuel remaining statistics, but
it gives him a graphical display of his distance from the lunar surface.
Give your friends a try at this one, then try to keep them away.

CRAPS
contributed by yours truly

I dug this one out of my high school senior scrapbook. It was a
project for my first ever computer class. I wrote it three years
ago, but have since refined it somewhat.

As with all the programs I present, I like to bring in something
new if at all possible. This program uses the statement PRINT CHR$(X)
to control the position of the cursor. The following table gives the
various values to move the cursor:

```
         1 - cursor left
        12 - clear screen
        17 - cursor home
        19 - cursor right
        23 - cursor up
        26 - cursor down
```

You will probably want to use the controls with PRINT CHR$(X);.
Note that semi-colon! It is not a typo, and if you leave it off when
trying to position the cursor (right, left, up, down) you may not get

the results you want. I'll leave you with this to play around with and learn about on your own.

MORE MISCELLANEOUS

Otto Borufsen of Sor-Audnedal, Norway, is our first international member. Congratulations, Otto.

I am working on some hardware-software designs for a beeper, a pair of joysticks, and a light pen. I will print schematics and full info as soon as any of them are finished. Hopefully, this will be in the next issue.

Kilobaud-Microcomputing has published a letter I sent them about seeking more members, and it is working. I am receiving several requests daily. My personal thanks to KB.

More material is needed for printing, as is almost always the case. Almost anything is acceptable: book reviews, hardware reviews, software, indeed, anything pertaining to the use of a Sorcerer. Let's all help each other!

This issue looks like about eight pages as of now. For now, I can and will absorb printing and special mailing costs, but if this newsletter grows the way I would like it to, I will have to start charging for subscriptions eventually. I would appreciate your views on this subject, i.e. what you think would be reasonable, should the format be changed to fit more on a page, should I take in advertisements, etc., etc.   I'd like to hear from you.

The following was contributed by Terry Calvert:
CROSS REFERENCE BETWEEN EXIDY SCRCERER AND TRS-80: Expansion Interface Jack

Function	Exidy Pin #	TRS-80 Pin #	Comment
DATA 0	37	30	
1	40	22	
2	39	32	
3	42	26	
4	41	18	
5	44	28	
6	43	24	
7	46	20	
ADDR 0	29	25	
1	32	27	
2	31	40	
3	34	34	
4	33	31	
5	36	35	
6	35	38	
7	38	36	
8	21	11	
9	24	17	
10	23	4	
11	26	9	
12	30	5	
13	27	6	
14	28	10	
15	25	7	
Clock 0 1	10		
0 2	13		
PRESET	1		
INT	2	14	TRS; INTAK= Interrupt Ack output
MI	16	21	INT = Interrupt input *Maskable
WAIT	3	33	Processor wait for memory
Data Buss	4		Exidy;
BUSS REQUEST	5		
NMI	6		Exidy; Nonmaskable interrupt
BUSS ACK	7		
BUSS DIRECTION	8	23	determines input/output direction
RAM CONTROL:			
RAM DRIVER	9		Exidy; also RCM ENABLE
REFRESH	19		
RD	17	15	Memory Read Strobe
WR	20	13	Memory write strobe
MUX		16	Multiplexor control for dynamic Rams
RAS		1	Row address strobe " " "
CAS		3	Column address strobe " " "
M REQ	15		
U.P. 8K	14		
RCM PRE	11		
I/O	48	12	TRS; CUT= Peripheral write strobe out
		19	IN = Peripheral read strobe out
I/O REQUEST	18		
RESET	45	2	Exidy; Pin 12= RESET ACK
HALT	22		
+5V.		39	
GND	49,50	8,29,37	
Unused	47		

A SHORT TOUR OF BASIC--ERATA

22 Mar 79

p.7,line 23   Add sentence: [ESC] by itself works the same as [RUN/STOP].

p.13,line 5   Replace: "A=A+1"

By: "A is equal to A+1".

p.60,line 15 Replace: ...to stimulate dice.

By: ...to simulate dice.

p.64,line 1   Add line:

          60 PRINT B$

p.69   Add section: String Space
              (see below)

p.71,lines   Replace: ORANGE$
      8 and 9

By: URANGE$

Note that ORANGE$ is not a legal name, since it contains the reserved word OR.

p.A-1   Add: RIGHT$

              STEP

after   Add Appendix H: The function USR(X)
p.G-2   (see below)

(L2)   String space

String space is the total amount of memory which Standard BASIC reserves for strings, string variables, and string arrays. Normally this is 50 bytes (characters); if the total number of characters in all your string constants, variables, and arrays is more then 50, you will get an OS error message. (In fact, you will get the OS message even sooner, since Standard BASIC uses some of the string space for its own purposes.)

PART 2

Example:

```
10 FOR X=1 TO 100

20 PRINT X,

30 LET A$=A$+"A"

40 PRINT A$

50 NEXT
```

This program will produce an OS error at X=26.

To reserve more string space, use the command

CLEAR <numerical expression>.

When Sorcerer sees this command, it first evaluates the expression and then reserves string space.

Examples:

This command	produces	this much string space
CLEAR 100		100 bytes
CLEAR 999		999 bytes
CLEAR X+Y		the number of spaces equal to the value of X+Y.

PROGRAM LISTING

```
1    REM ** RACE TRACK **
2    REM ** M. WEINGAST **
5      PRINT CHR$(12)
10   FOR X=5 TO 35
20     POKE -3000+X,177
30     NEXT X
100    FOR X=-7 TO 7
110      POKE -2965+64*X,177
120      POKE -2995+64*X,177
130      NEXT X
200    FOR X=0 TO 6
210      POKE -3173-64*X,177
220      POKE -2788+64*X,177
230      NEXT X
300    FOR X=-10 TO 10
310      POKE -3000+64*X,177
320      POKE -2960+64*X,177
330      NEXT X
400    FOR X=0 TO 40
410      POKE -2360+X,177
420      POKE -3640+X,177
430      NEXT X
500 Z=-2999
505    T$=CHR$(48)
```

```
510    DATA   1,205,2,9,3,224,4,50,5,0,6,0,7,201,260,1,261,c
520    FOR  I=1 TO 9
521    READ A,B
522    POKE A,B
523    NEXT I

600    FOR S=0 TO 1000
610    D5 = USR(D5): W$ = CHR$(PEEK(8))
615    IF W$ <> CHR$(0) THEN T$=W$
620    B= ASC(T$)-48

700    IF B=8 THEN P=-64
710    IF B=6 THEN P=1
720    IF B=4 THEN P=-1
730    IF B=2 THEN P=64
740    IF B=5 THEN P=0

800    Z=Z+P
850    POKE Z,42
860    FOR A=0 TO 10: NEXT A
870    POKE Z,32
880    FOR X=0 TO 100: NEXT X
900    NEXT S
1000   END
```

```
10 PRINTCHR$(12)
20 PRINT"--------- LUNAR LANDING SIMULATION ----------"
30 PRINT:PRINT
40 PRINT:INPUT"DO YOU WANT INSTRUCTONS";Q$:PRINT
50 IF Q$="Y" THEN 270
60 PRINT
70 PRINT
80 PRINT "YOU ARE LANDING ON THE MOON AND HAVE TAKEN OVER MANUAL
"
90 PRINT"CONTROL 500 FEET ABOVE A GOOD LANDING SPOT. YOU HAVE A"
100 PRINT"DOWNWARD VELOCITY OF 50 FT/SEC. 150 UNITS OF FUEL REMA
IN."
110 PRINT
120 PRINT"HERE ARE THE RULES THAT GOVERN YOUR SPACE VEHICLE:"
130 PRINT"(1) AFTER EACH SECOND, THE HEIGHT, VELOCITY, AND REMAI
NING"
140 PRINT"     FUEL WILL BE REPORTED."
150 PRINT"(2) AFTER THE REPORT, A '?'WILL BE TYPED.   ENTER THE"
160 PRINT"     NUMBER OF UNITS OF FUEL YOU WISH TO BURN DURING TH
E"
170 PRINT"     NEXT SECOND. EACH UNIT OF FUEL WILL SLOW YOUR DESC
ENT"
180 PRINT"     BY 1 FT/SEC."
190 PRINT"(3) THE MAXIMUM THRUST OF YOUR ENGINE IS 30 FT/SEC/SEC
OR"
200 PRINT"     30 UNITS OF FUEL PER SECOND."
210 PRINT"(4) WHEN YOU CONTACT THE LUNAR SURFACE, YOUR ENGINE"
220 PRINT"     WILL AUTOMATICALLY CUT OFF AND YOU WILL BE GIVEN A
"
230 PRINT"     REPORT OF YOU LANDING SPEED AND REMANING FUEL."
240 PRINT"(5) IF YOU RUN OUT OF FUEL, THE '?' WILL NO LONGER APP
EAR,"
250 PRINT"     BUT YOUR SECOND-BY-SECOND REPORT WILL CONTINUE UNT
IL"
260 PRINT"     YOU CONTACT THE LUNAR SURFACE.": PRINT
270 INPUT"PUSH RETURN WHEN READY";Q$
280 PRINTCHR$(12)
290 PRINT"BEGINNING LANDING PROCEDURE.....": PRNT
300 PRINT"G O O D   L U C K ! ! !"
310 PRINT
320 PRINT"SEC  FEET    SPEED     FUEL        PLOT OF DISTANCE"
330 PRINT
340 T=0
350 H=500
360 V=50
370 F=150
380 PRINT T;TAB(4);H;TAB(12);V;TAB(20);F;
390 PRINT TAB(25);CHR$(168);
400 PRINT TAB(H/15+26);CHR$(159);CHR$(158)
```

```
410 INPUTB
420 IF B<0 THEN 570
430 IF B>30 THEN B=30
440 IF B>F THEN B=F
450 V1=V-B+5
460 F=F-B
470 H=H-.5*(V+V1)
480 IF H<=0 THEN 590
490 T=T+1
500 V=V1
510 IF F>0 THEN 380
520 IF B=0 THEN 540
530 PRINT"****OUT OF FUEL****"
540 PRINT T;TAB(4);H;TAB(12);V;TAB(20);F;
550 PRINT TAB(25);CHR$(163);
560 PRINT TAB(H/15+26);CHR$(159);CHR$(158)
570 B=0
580 GOTO 450
590 PRINT"****CONTACT****"
600 H=H+.5*(V+V1)
610 IF B=5 THEN 640
620 D=(-V+SQR(V*V+4*(10-2*B)))/(5-B)
630 GOTO 650
640 D=H/V
650 V1=V+(5-B)*D
660 PRINT"TOUCHDOWN AT";T+D;"SECONDS."
670 PRINT"LANDING VELOCITY=";V1;"FEET/SEC."
680 PRINTF;"UNITS OF FUEL REMAINING."
690 IF V1<>0 THEN 730
700 PRINT"CONGRATULATIONS!  A PERFECT LANDING!"
710 PRINT"YOUR LICENSE WILL BE RENEWED.........LATER"
720 GOTO 760
730 IF ABS(V1)<2.5 THEN 840
740 PRINT"******SORRY, BUT YOU BLEW IT!!!!!"
750 PRINT"CONDOLENCES TO YOUR NEXT OF KIN"
760 PRINT:PRINT:PRINT
770 PRINT"ANOTHER MISSION? (Y/N)
780 INPUT Z$
790 IF Z$="N" THEN 810
800 GOTO 230
810 PRINT"CONTROL CENTRAL, OUT, SO LONG"
820 PRINT:PRINT:PRINT:PRINT
830 GOTO 870
840 PRINT"TOUCH DOWN WAS A LITTLE ROUGH, BUT YOU SURVIVED!!"
850 PRINT:PRINT:PRINT
860 GOTO 770
870 END
READY
```

```
10 REM                    CRAPS BY DAVE BRISTOR
20 REM WRITTEN SPRING 1976
30 REM REVISED SUMMER 1979
40 REM INITIALIZE RANDOM NUMBER GENERATOR
50 D=RND(-RND(5))
60 PRINT CHR$(12)
70 PRINT:PRINT"                        ***CRAPS***
80 PRINT:PRINT"DO YOU NEED INSTRUCTIONS ";
90 INPUT A$:IF LEFT$(A$,1)<>"N" THEN GOSUB630   :GOTO110
100 PRINT:PRINT"OKAY THEN, I'M GAME.
110 C=1:M=0
120 REM INPUT LOOP TO LINE 190
130 PRINT "TYPE ROLL ";
140 INPUT R$:IF R$="ROLL" THEN M=0:GOTO210
150 M=M+1
160 ON M GOTO170 ,180,190
170 PRINT:PRINT"PLEASE ROLL "::GOTO140
180 PRINT:PRINT"I SAID ROLL!!! ";:GOTO140
190 PRINT:PRINT"OKAY THEN, I QUIT!!!!!!":GOTO790
200 REM ROLL DICE
210 X=INT(RND(1)*6+1)
220 Y=INT(RND(1)*6+1)
230 S=X+Y
240 IF C=1 THEN FST=S
250 GOSUB340
260 REM         CHECK FOR WIN
270 IF C=1 AND (S=7 OR S=11) THEN400
280 IF C>1 AND S=FST THEN400
290 REM         CHECK FOR LOSE
300 IF C=1 AND (S=2 OR S=3 OR S=12) THEN470
310 IF C>1 AND S=7 THEN470
320 C=C+1
330 GOTO130
340 REM         PRINT ROLL SUBROUTINE
350 IF S=8 OR S=11 THEN GOTO380
360 PRINT:PRINT"THAT'S A";X;"AND A";Y;", WHICH MAKES FOR A";S
370 GOTO390
380 PRINT:PRINT"LOOKS LIKE A";X;"AND A";Y;"-A TOTAL OF";S
390 RETURN
400 REM         ***WIN***
410 PRINT:PRINT"            YOU WON!!!!
420 IF C<3 THEN PRINT:PRINT"LUCKY FINGERS, DUDE":GOTO450
430 IF C<6 THEN PRINT:PRINT"NOT BAD FOR AN AMATEUR":GOTO450
440 PRINT:PRINT"WELL FINALLY!  AT LEAST YOU WON.":GOTO450
450 PRINT:PRINT"YOU HIT IT LUCKY ON ROLL NUMBER";C
460 GOTO530
```

```
470 REM          **LOST***
480 PRINT:PRINT"        AHA-YOU LOSE!
490 IF C<3 THEN PRINT:PRINT"THAT DIDN'T TAKE LONG, DID IT ?":GOT
0520
500 IF C<6 THEN PRINT:PRINT"KEEP UP THE GOOD WORK, PAL":GOTO520
510 PRINT:PRINT"DOWN THE DRAIN AFTER ALL THAT!":GOTO520
520 PRINT:PRINT"YOU BLEW IT ON ROLL";C
530 REM          ***ROLL NUMBER***
540 IF C<3 THEN PRINT:PRINT"TRY AGAIN, HOTSHOT "::GOTO570
550 IF C<6 THEN PRINT:PRINT"WHY NOT GIVE IT ANOTHER TRY "::GOTO5
70
560 PRINT:PRINT"ONCE MORE "::GOTO570
570 INPUT A$
580 IF LEFT$(A$,1)<>"N" THEN110
590 PRINT:PRINT"OKAY, THANKS FOR PLAYING. BYE!
600 PRINT
610 END
620 REM               INSTRUCTIONS
630 PRINT CHR$(12)
640 PRINT
650 PRINT" CRAPS IS A DICE GAME.  YOU WIN OR LOSE DEPENDING
660 PRINT"ON WHAT YOU ROLL.
670 PRINT
680 PRINT" TO WIN, JUST ROLL A 7 OR AN 11 ON YOUR FIRST
690 PRINT"ROLL (HA!), OR ROLL YOUR FIRST NUMBER ON ANY
700 PRINT"SUCCEEDING ROLL.
710 PRINT
720 PRINT" TO LOSE, MAKE YOUR FIRST ROLL A 2, A 3, OR A 12.
730 PRINT"OR, YOU CAN LOSE BY ROLLING A 7 ANY TIME AFTER
740 PRINT"YOUR FIRST ROLL.  NOW ISN'T THAT EASY ?
750 PRINT
760 PRINT" GET READY NOW, 'CAUSE HERE WE GO...
770 PRINT
780 RETURN
790 END
READY
```